MEMORIA PFC

Implementación de algoritmos de enrutamiento para redes inalámbricas de sensores móviles

DBR: DEPTH-BASED ROUTING PROTOCOL

PARA REDES DE SENSORES SUBACUÁTICOS

Escuela Técnica Superior de Ingeniería Informática etsinf

Septiembre 2011

Autor: Daniel Mazón Menéndez

Directores: Salvador Climent Bayarri, Juan Vicente Capella Hernández

Índice

1. Introducción

Se va a proceder a estudiar la viabilidad y la capacidad de un algoritmo de enrutamiento para redes de sensores subacuáticos. Si bien es cierto que existen un gran número de algoritmos de enrutamiento para redes de sensores móviles, éstos están construidos sobre una arquitectura física completamente distinta, basada en ondas electromagnéticas. Se hace, pues necesaria, la adaptación, o creación desde cero, de un método de encaminamiento para las redes subacuáticas, que tenga en cuenta las peculiaridades de este medio, destacando especialmente las siguientes: la velocidad de propagación enormemente inferior (en torno a 5 órdenes de magnitud), un rango de transmisión generalmente menor al que ofrecen las ondas electromagnéticas en el aire, aunque dependiendo de la potencia y la frecuencia se puede transmitir bastante lejos; además también de la velocidad de transmisión, de una capacidad también reducida. Estas diferencias vienen dadas por el hecho de que en vez de utilizar ondas de radiofrecuencia, que sufren una atenuación muy fuerte en el agua, lo que conlleva un radio de acción muy corto, se utilizan ondas acústicas, ondas de sonido, de forma similar al funcionamiento de un sónar. Las frecuencias usadas en este tipo de sistemas van desde los 20 Hz hasta los 20 kHz (el rango del oído humano), sin bien es cierto que existen otros de alta frecuencia (85 a 350 kHz), aunque cuentan con un alcance más reducido. [3]

Las redes de sensores subacuáticos son un campo de importancia para diversos campos, por los datos que pueden aportar en diversas áreas de investigación. Por ejemplo, una cantidad de yacimientos arqueológicos de importancia son encontrado actualmente en zonas inundadas o hundidas en el mar, por lo que se hace necesario una forma de explorarlos. También en geología es de importancia por la cantidad de datos que puede aportar sobre tectónica placas, movimientos de tierra y previsión de fenómenos geológicos de alta violencia como erupciones volcánicas. Las redes subacuáticas también permiten el estudio del ecosistema marino y las especies que en él habitan, en donde la compresión biológica del entorno permite una mejor gestión de la capacidad pesquera de la zona. También cabría mencionar que existen actividades militares interesadas en este campo.

Se trata de un área de investigación de reciente aparición, en la que los dispositivos físicos reales son escasos y caros, además de las dificultades que supone el despliegue real de un banco de pruebas, por lo que se hace necesario la utilización de un medio de simulación que permita poner a prueba los algoritmos propuestos de una forma realista y eficiente. En un posterior despliegue real, las simulaciones nos permitirán ajustar lo mejor posible los parámetros del protocolo estudiado con un coste muchísimo menor que el que supondría probarlo directamente en la realidad. Para que esto sea posible, es necesario que el simulador sea lo más realista posible en sus resultados.

En nuestro caso, emplearemos la suite académica ns-3 (network simulator versión 3), heredera de la ampliamente conocida ns-2. En la versión actual, aparecen ya implementados las capas física y MAC de la pila de protocolos para las redes subacuáticas. Se incluyen varios modelos de propagación, de cálculo de ruidos, de medios físicos así como de protocolos de acceso al medio. También cabe destacar que existen otros elementos del simulador que nos serán de gran ayuda en el desarrollo del proyecto: sistemas de posicionamiento, para situar los nodos en un espacio tridimensional, sistemas de movilidad, para simular que los nodos no permanecen quietos en el mismo punto o sistemas de alimentación energética, para simular el consumo energético y eventual desconexión de los nodos que agoten sus reservas.

Por lo tanto vamos a ocuparnos de crear un nuevo elemento sobre una capa de la pila de protocolos aún no estudiada para el simulador ns-3: vamos a implementar un protocolo en el nivel de red para

sensores inalámbricos. Entre las varias alternativas posibles, se elige escribir uno pensado especialmente para este tipo de dispositivos, y aún más, en una configuración concreta. Es el protocolo DBR, por las siglas en inglés Depth Based Routing Protocol, o protocolo de enrutamiento basado en la profundidad.

Este protocolo asume que se está generando un flujo de datos unidireccional, procedente de unos nodos con sensores hacia uno o varios sumideros que recoge la información enviada, y situados generalmente en la superficie del agua. Es un protocolo posicional: se asume que los nodos disponen de algún dispositivo que les permite conocer a qué profundidad se encuentran, lo cual es un prerrequisito realista, pues sensor de presión nos daría este dato con una construcción fácil y barata, en comparación con los protocolos posicionales que asumen que los nodos pueden conocer su posición en los 3 ejes dimensionales.

Por tanto, durante el funcionamiento de una red de estos sensores, los nodos irán pasándose la información desde los más profundos a los menos, hasta llegar al sumidero situado en la superficie. Este esquema da mucho juego para optimizar diversos parámetros que permitan una mayor duración de la batería de los nodos, un menor retardo o una mayor tasa de entrega de paquetes. [1]

1.1 Objetivos

Este proyecto tiene como primer objetivo la implementación del protocolo de enrutamiento DBR en el simulador ns-3. Una vez implementado, perseguimos hacer un estudio intensivo de éste, comparándolo en primer lugar con los resultados del artículo que tomaremos como referencia, y en segundo lugar, desarrollando nuevos tests para cubrir casos no tratados, y en especial, el caso del modo extendido que introducimos como novedad respecto al artículo.

Esto nos permitirá también el estudio en profundidad del módulo para redes subacuáticas del simulador ns-3. Actualmente hay implementados los niveles inferiores de la pila de protocolos: la física y la MAC. La primera incluye el canal, modelos de propagación, de SINR, etc... y la segunda varios propuestas distintas entre las que elegir. De esta forma, nuestro protocolo añadirá una nueva capa aún no explorada en la suite.

1.2 Repaso de protocolos de enrutamiento

Existen principalmente dos grandes familias de protocolos de enrutamiento para redes ad hoc: los reactivos y los proactivos.

Los proactivos son algoritmos en los que los nodos mantienen una serie de tablas con información de rutas al resto de nodos de la red (o al menos a una parte de ésta). Los nodos han de encargarse de mantener actualizada esta información, por si la topología de la red cambia. Un ejemplo de este tipo es el algoritmo DSDV (Destination-Sequenced Distance Vector), que tiene 2 tipos de paquetes para mantener actualizada la información de rutas “full dump” e “incremental”, el primero conteniendo toda la información y el segundo únicamente las diferencias respecto al último “full dump”. Sin

embargo, tiene el problema de introducir una gran cantidad de *overhead* debido a las actualizaciones periódicas necesarias.

Los reactivos son algoritmos diseñados para reducir el problema del *overhead* de los proactivos, haciendo que los nodos únicamente guarden información de las rutas activas, y por tanto las rutas sean determinadas únicamente cuando hay datos a enviar, proceso que se hace normalmente por peticiones en *flood* hasta dar con el destino. Un ejemplo de este tipo de protocolos es el algoritmo AODV (Ad hoc On-demand Distance Vector), en donde los nodos envían paquetes *hello* para colaborar en la búsqueda de rutas y en donde los paquetes llevan como cabecera la información del nodo de destino. [5]

Existen algunos protocolos diseñados específicamente para redes subacuáticas, como VBF (Vector-based forwarding protocol), pero tienen la desventaja de ser protocolos geográficos, es decir, que necesitan tener información de dónde está situado el nodo en las 3 dimensiones para poder reenviar el paquete, una dificultad no trivial de resolver. [7]

2. Descripción del caso de uso

El protocolo elegido está pensado para obtener el mejor rendimiento en un caso de uso concreto: un reducido grupo de nodos sumideros situados en la superficie, y un grupo de nodos fuente que generan y envían los datos. Puede colocarse un número variable de nodos intermedios que se encargarán de retransmitir los paquetes desde su origen hasta su destino. Nótese que el flujo de datos es unidireccional.

Generalmente se asume que existen 2 tipos de protocolos de enrutamiento para redes adhoc: proactivos y reactivos. Los primeros buscan y guardan la información de rutas por sí mismos para cada nodo, mientras que los segundos únicamente calculan las rutas necesarias para los paquetes que han de enviarse en cada momento. Podríamos clasificar DBR como reactivo, pues no guarda información de rutas, sino que toma una decisión para cada paquete que le llega.

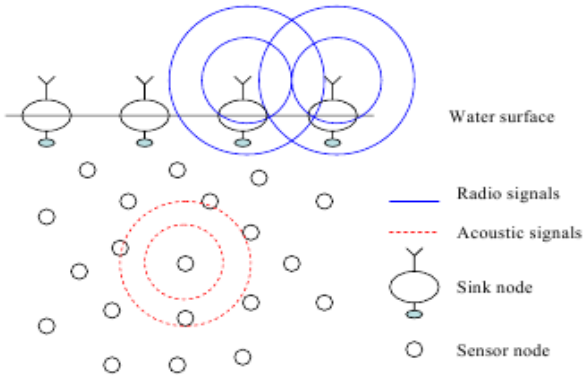

Imagen 1

2.1 Descripción del protocolo

El protocolo, al tener que entregar los nodos únicamente a los sumideros, sin tener que preocuparse de devolver los datos a los nodos fuentes, está pensado para comportarse de forma óptima en este sentido, pues los paquetes siempre fluirán desde una profundidad mayor a una menor. En este sentido, se hace necesario añadir un campo en la cabecera del paquete que identifique de que profundidad viene, para que cada nodo pueda tomar la decisión de reenviarlo o no. Si el nodo que lo recibe está a mayor profundidad, descartará el paquete en todo caso, pero si por el contrario, se encuentra más cerca de la superficie que el nodo del cual procede el paquete, se convertirá en un candidato para la retransmisión. Cada vez que un nodo retransmite un paquete, actualizará el campo de profundidad de la cabecera al valor que tenga registrado en ese momento para su sensor de profundidad.

Para optimizar el tiempo de batería de los nodos, conviene reducir el número de nodos intermedios que retransmitirán los datos. Para ello, cuando un nodo recibe un paquete, y es candidato a reenviarlo por encontrarse a menor profundidad, no procederá de inmediato a ello, sino que esperará un tiempo prudencial para dar la posibilidad a otros nodos, aún a menos profundidad, a que también escuchen y reenvíen los datos, ahorrándose de esta forma el tener que realizarlo, con el consumo energético asociado a ello.

Se definirá asimismo un umbral mínimo para retransmitir paquetes, y se creará una cola de enviados donde se guardarán los datos identificativos de los paquetes para evitar múltiples retransmisiones, con el consumo de energía asociado. Algunos paquetes escuchados pero no retransmitidos también se guardarán en esta cola para evitar su posible reenvío posterior. Todo esto queda esquematizado en el diagrama 1.

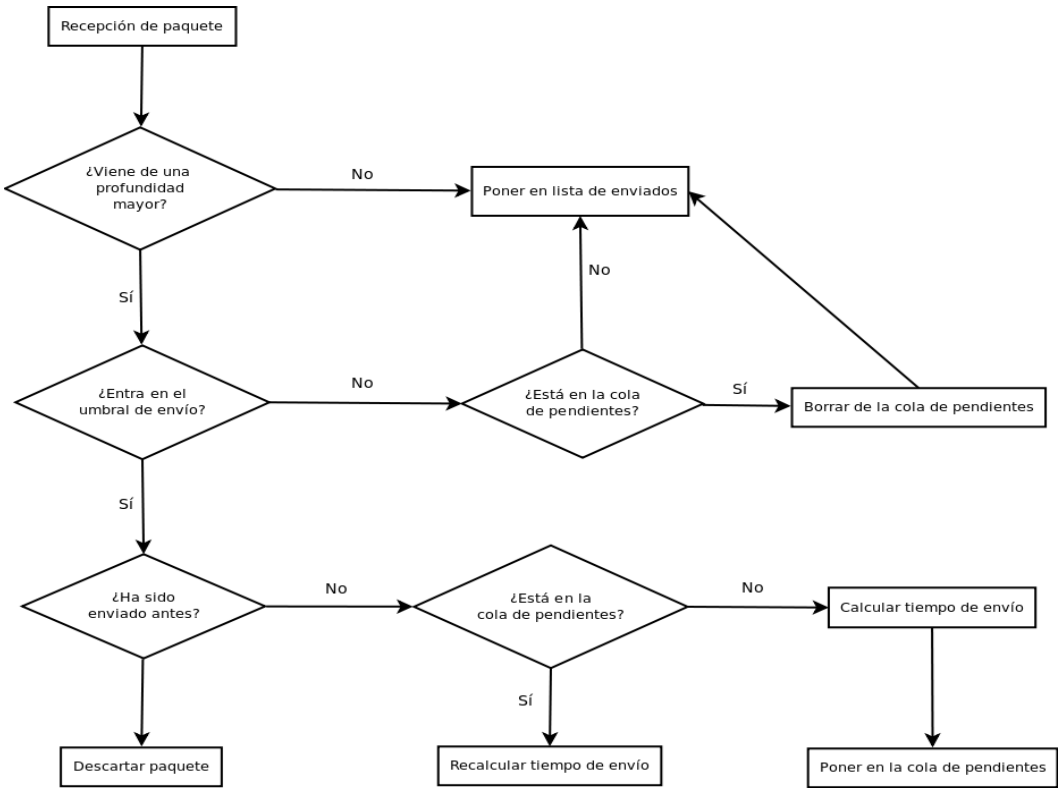

Diagrama 1

El tiempo que un nodo espera entre que recibe un paquete y empieza a reenviarlo se llama Holding Time. La forma de calcularlo resultará crucial para ofrecer unos buenos resultados. Lo ideal sería que el candidato diera tiempo suficiente para que todos los nodos dentro del radio de alcance del transmisor original pudieran escuchar el paquete, y que los situados en el punto a menor profundidad lo retransmitieran de vuelta hasta la posición del candidato. Esto puede resolverse mediante un sistema de ecuaciones en donde tendremos que definir un radio de alcance máximo, que puede saberse a priori de forma sencilla, la profundidad del nodo emisor, contenida en la cabecera, y la del nodo candidato, que puede saberse a partir de su propio sensor, y algunos datos extras constantes como la velocidad de propagación del sonido en el agua, de 1500 m/s.

Se necesitan pues que el cálculo del Holding Time satisfaga las siguientes condiciones: que los nodos a menor profundidad tengan tiempos menores, y que el tiempo de espera para un nodo sea suficiente como para permitir que el paquete llegue hasta el límite del radio de propagación y sea retransmitido de vuelta por un nodo situado en ese punto. Para definir unas ecuaciones que cumplan estas condiciones se hace necesaria la introducción de 2 parámetros variables: **delta**, que sustituirá la distancia mínima entre dos nodos candidatos dentro del radio de transmisión, y el **umbral** mínimo de diferencia de profundidad respecto al salto anterior, por debajo del cuál un nodo no retransmitirá ningún paquete, aunque se encuentre a una profundidad menor. Estos parámetros nos permitirán ajustar con mayor o menor precisión el tiempo de espera, y ver cómo varían los resultados de la simulación a medida que jugamos con ellos.

Para que esto funcione, un nodo ha de ser capaz de identificar de forma única cada paquete, para no retransmitirlo de nuevo si lo escucha de un nodo superior. Para ello, en la cabecera añadiremos 2 campos más, un ID de nodo de origen, que en nuestro caso no corresponderá a la dirección del nodo en que se generó originalmente el paquete, como explicaremos más adelante, y un número de secuencia, para diferenciar los distintos paquetes provenientes del mismo nodo. A diferencia del campo de profundidad, estos dos campos no serán modificados por ningún nodo retransmisor intermedio.

$$f(d)=\frac{2\tau}{\delta}\cdot(R-d),\delta\in(0,R]$$

Ecuación 1

De esta forma, cada nodo deberá guardar en una memoria interna 2 estructuras de datos: por un lado una cola de paquetes pendientes de envío, y también una lista de paquetes ya enviados, para no reenviarlos de nuevo si vuelve a escucharlos. Si durante el holding time, se recibe desde un nodo a menor profundidad un paquete que se encuentra en la cola de pendientes, éste pasará a la lista de enviados automáticamente sin ser retransmitido. También se introducirán en la lista de enviados los paquetes que se reciban desde una profundidad mayor, y por supuesto, aquellos que sean retransmitidos por el nodo en cuestión.

En la ecuación 1 tenemos la fórmula que obtiene el tiempo de espera para un nodo situado a una distancia d del nodo anterior. Se calcula a partir de τ, que es el tiempo máximo de propagación para el rango máximo de propagación, R. También tenemos el parámetro δ, que permite ajustar la diferencia mínima para retransmitir un paquete, y cuyo impacto en los resultados estudiaremos en profundidad más adelante.

2.2 Modo extendido

La descripción del protocolo proporcionada por el paper no tiene en cuenta el tiempo de transmisión para el cálculo del Holding Time, únicamente el de propagación. Para velocidades de transmisión altas, se puede obviar este dato, ya que se traduce en un tiempo muy pequeño que no da espacio a causar colisiones. Sin embargo, para velocidades reducidas, como las que podemos encontrar en

dispositivos acústicos, se vuelve un parámetro crítico. Por ello, se ha implementado un modo extendido que tiene en cuenta este dato para el cálculo del Holding Time, y que puede activarse o desactivarse fácilmente por parámetro.

La ecuación 2 es igual que la 1 pero añadiendo la variable tr, que representa el tiempo de transmisión del paquete actual.

$$f(d)=\frac{2\tau+tr}{\delta}\cdot(R-d),\delta\in(0,R]$$

Ecuación 2

3. Implementación

3.1 Descripción del simulador ns-3

Para la implementación del protocolo vamos a utilizar el simulador ns-3, concretamente en su versión 3.10. Se trata de un simulador de eventos discretos, usado ampliamente por la comunidad académica e investigadora, y de hecho es muy utilizado para el estudio de redes ad hoc.

Está escrito en C++ y cuenta con una gran diversidad de módulos, que permiten modelar diversos aspectos interesantes a estudiar: la movilidad y distribución espacial de los nodos, su consumo energético y por supuesto, el de mayor importancia para nosotros: la capa de red física y MAC para redes subacuáticas.

La versión actual es un fork de la antigua versión ns-2, que estaba escrita en C y usaba un lenguaje de scripting basado en Tcl para definir los tests. En 2005 se empezó a desarrollar la nueva versión ns-3 desde cero y basada exclusivamente en C++, en la que se aprovechó para solventar algunas de las deficiencias de la versión anterior, especialmente las referidas al lenguaje de scripting y la falta de modularización. En 2008 se lanzó la primera versión de esta rama. [6]

3.2 Desarrollo del protocolo

Se han implementado dos clases principales para dar forma al protocolo. Por un lado la clase DBRHeader() y por otro la clase DBRNode(). El diagrama 2 muestra un esquema de las relaciones y elementos más importantes de estas clases.

La clase DBRHeader contiene los métodos necesarios para crear las cabeceras que se añadirán a los paquetes. A cada paquete enviado deberá añadírsele dicha cabecera para ser manejada por el protocolo y que éste pueda tomar las decisiones correctas de retransmisión de datos. Esta clase se encargará de serializar y deserializar los campos de la cabecera. Para la codificación de éstos, se utiliza un tipo float para la profundidad, un tipo uint16_t para el ID del nodo origen, y un tipo uint16_t para el número de secuencia. Todo ello suma un tamaño total de la cabecera de 8 bytes. Podría utilizarse un tipo de menor tamaño para el número de secuencia, dependiendo de la cantidad de paquetes que vayan a ser enviados por unidad de tiempo y de la duración de los datos en la caché de paquetes enviados de los nodos. No se utiliza un tipo uint8_t para la dirección de origen de los paquetes (que es el tamaño con que se almacena dentro de la clase UanAddress ya implementada en el ns-3) pues limita el número de nodos a 255, lo cual introducirá algunas limitaciones para realizar simulaciones con números de nodos superiores, tal como propone el paper.

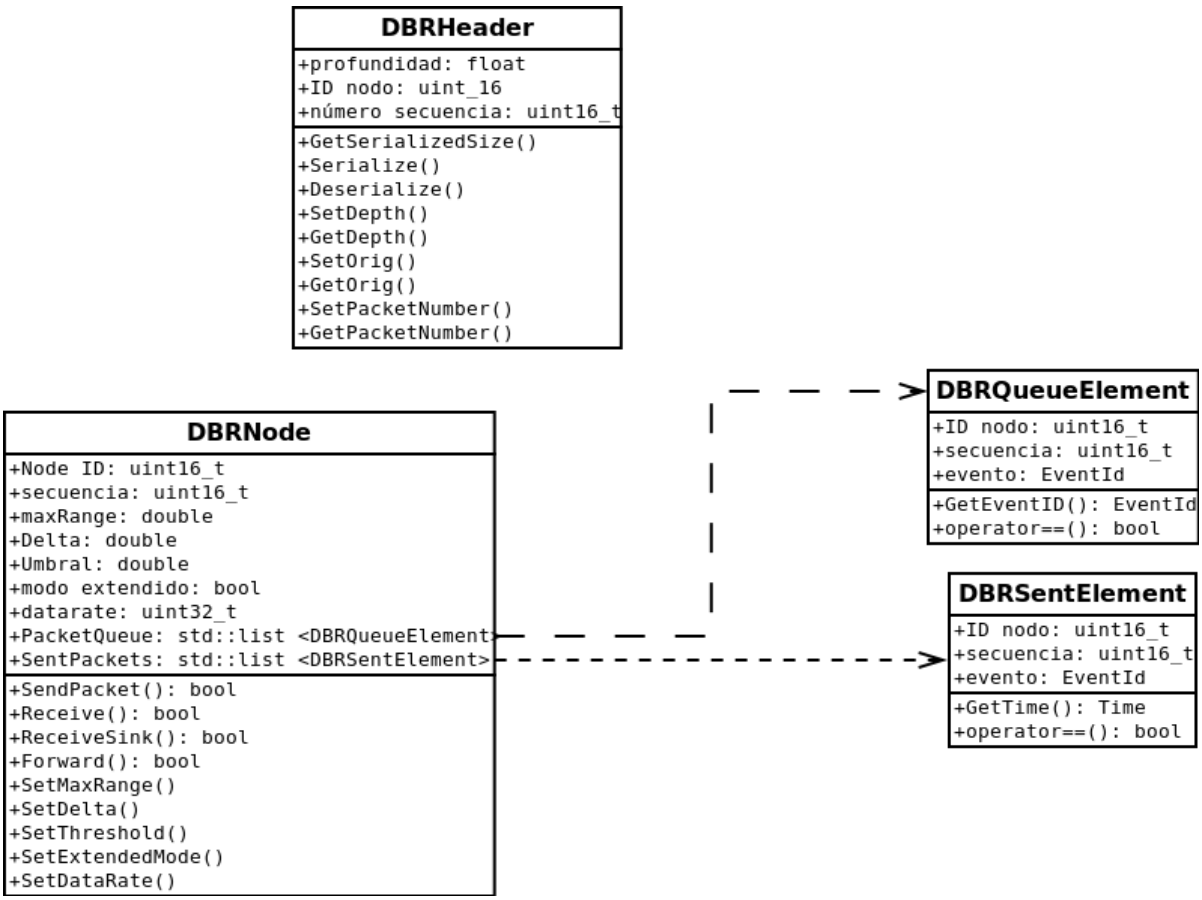

Diagrama 2

Por otro lado se ha creado la clase DBRNode que se instalará en los nodos que deban tratar con este protocolo. Tiene dos funciones principales: Receive() y Send().

Receive() se encargará de recibir los paquetes que lleguen hasta el nodo, y será invocado por las capas inferiores de la pila de protocolos. Tomará una primera decisión sobre descartarlo o prepararlo para ser retransmitido.

La función de Send() se encargará de llamar a las capas inferiores con el paquete que haya de ser enviado, ya con su cabecera correspondiente debidamente insertada.

Otro método de gran importancia es Forward(), que se encargará de recibir los paquetes aceptados para el reenvío por Receive() y calcular su Holding Time y programarlos para ser enviados.

3.3 Descripción de la cabecera

La cabecera que se añade a cada paquete es de vital importancia para el correcto funcionamiento del protocolo. Ha sido implementada mediante la clase DBRHeader, y consta de tres campos:

– Deep: El campo que trata la profundidad desde la que se emite cada paquete. Es actualizado en cada salto y se codifica mediante un float de 32 bits. Otras alternativas estudiadas serían un double de 64 bits, pero aumentaría considerablemente el tamaño requerido sin añadir realmente un precisión útil. En un caso de uso extremo, las zonas más profundas del océano alcanzan los 11.000 metros [2], dato que puede codificarse sin problemas en 32 bits, manteniendo una precisión mejor que un metro a cualquier profundidad, cuando seguramente ningún sensor disponible en el mercado llegue a tanto. Se prefiere un tipo de datos que permita almacenar número reales frente a enteros, aunque podría argumentarse que los sensores disponibles para medir la profundidad no tienen la precisión suficiente para medir distancias inferiores a un metro, especialmente a grandes profundidades, por lo que un entero de 16 bits sería suficiente para dicho campo, además de sensiblemente más pequeño, a lo que hay que añadir que los parámetros de delta y umbral harían estas mediciones poco relevantes. Sin embargo, se prefiere tomar la decisión de contar con esta capacidad.

– NodeID: Es el identificador del nodo, que se almacena como un entero sin signo de 16 bits. Sirve para identificar de forma única a cada nodo. No es la dirección UAN, aunque cumple una función similar. No se ha utilizado este último tipo, pues se codifica como un entero de 8 bits, lo que reduce el número de posibles nodos a 255, impidiéndonos realizar simulaciones con número mayores tal como está en el paper. Por ello empleamos 16 bits, lo que nos proporciona más de 65000 posibles nodos, que estimamos un tamaño suficiente para nuestras simulaciones e hipotéticos despliegues reales. Al instalar el nodo, se le ha de asignar un valor.

– Sequence number: Es el número de secuencia, para identificar de forma única a cada paquete de un mismo nodos. Hemos utilizado igualmente un entero sin signo de 16 bits. Dependiendo del número máximo de envíos por unidad de tiempo y de la duración de la cache de paquetes enviados, podría reducirse a un entero de 8 bits, pero para evitar posibles inconvenientes elegimos uno de 16. Se inicializa a 0 al arrancar, y se incremente en uno con cada paquete enviado.

4. Experimentación

4.1 Parámetros utilizados

Para configurar las capa física de la red UAN se utiliza el método estándar proporcionado por el simulador para el cálculo del SINR (ratio señal-interferencia), así como el método estándar de cálculo de errores en los paquetes.

A ello, hay que sumar que se utiliza como modulación FSK (modulación por desplazamiento de frecuencias) , con una frecuencia central de 85kHz, un ancho de banda de 1kHz y una constelación de 2 valores. Normalmente se usa a un datarate de 1500 bits/s, aunque realizaremos varias pruebas cambiando este valor para probar el modo extendido del protocolo.

Hay que añadir que la potencia de emisión se sitúa en 190 dB, lo que permitirá un radio de alcance de unos 100 metros. Todo ello se combina junto con un modelo de propagación usando la conocida aproximación de Thorp (similar al modelo de canal implementado en ns-2) y con un canal adecuado para constituir la capa física de la red.

Para la capa MAC (de control de acceso al medio) disponemos de 3 alternativas: Aloha, CW y RC: Aloha es el protocolo utilizado por ser el más similar al descrito en el paper. Aloha transmite nada más tiene un paquete que enviar, sin importar en que estado se encuentre el medio. El paper describe como MAC un método que escucha el medio antes de enviar, y si lo encuentra libre procede al envío. Si no, se espera un tiempo para volver a intentarlo, y el paquete se descarta si encuentra el medio ocupado un número determinado de veces.

Otra alternativa disponible es la MAC CW, que utiliza una ventana de contención dividida en slots, de tamaño configurable. Si el medio está ocupado, el nodo esperará un número aleatorio de slots antes de volver a intentarlo. El tamaño de los slots también es configurable. Sin embargo, éste método presenta el inconveniente de que una vez la capa superior le entrega un paquete para su envío, pierde todo control sobre éste: por lo que no hay forma de que, si alguien emite el mismo paquete desde un nodo superior entre el momento de la entrega del paquete a la MAC y el momento de su transmisión, sea descartado de la cola de envío, pues otro nodo ya se ha adelantado.

La tercera alternativa es la MAC RC, que utiliza un canal de datos y otro de control, pero que asume una red con una puerta de enlace y todos los nodos situados a un salto, por lo que no nos resulta de utilidad para nuestro caso de uso.

4.2 Distribución espacial

Los nodos se distribuyen en un espacio tridimensional con forma de cubo de 500 metros de lado. En la superficie se colocan los nodos sumideros, a una profundidad de 1 metro. Por convención, la profundidad viene dada por la coordenada Z, y se expresa como un valor negativo. De esta forma, Z=0 sería la superficie, -1 correspondería a un metro de profundidad, etc...

A ello se le ha añadido el esquema de movilidad de Gauss Markov, para trazar trayectorias aleatorias dentro de un espacio 3D, lo que nos permite simular el movimiento de los nodos. Existen varios parámetros para configurar la trayectoria que siguen los objetos, pero nosotros nos quedaremos principalmente con el que nos permite establecer el rango de velocidades con que se moverán los nodos.

Todos los nodos intermedios se distribuirán inicialmente mediante el RandomBoxPositionAllocator, y se les instalará el modelo de movilidad descrito. En cuanto al nodo fuente, se situará a 500 metros de profundidad, en el centro del cuadrado (x = 250, y = 250). Los nodos sumideros son 5 en total, y se sitúan a 1 metro de profundidad y unas coordenadas sobre el plano horizontal siguientes: (125, 125), (125, 375), (250, 250), (375, 125), (375, 375). Ni el nodo fuente, ni los nodos sumideros se mueven durante la simulación.

Finalmente, a los nodos móviles se les instalará una batería de una determinada capacidad configurable para simular el momento en que se agote su capacidad de transmisión. Los consumos de este dispositivo son los siguientes: 2 W para transmisión, 100 mW para recepción y 10 mW para idle.

4.3 Resultados

Habiendo ejecutado diversas simulaciones para los parámetros previamente descritos, enviando 300 paquetes de 50 bytes, uno por segundo, usando MAC Aloha, un datarate de 10 kbps, una velocidad de desplazamiento máxima de 1 m/s y una energía inicial de 500 J, obtenemos los siguientes resultados.

Las primeras simulaciones que vamos a realizar serán comparativas con las propuestas por el paper, para comprobar que nuestra implementación arroja resultados parecidas.

En primer lugar, estudiaremos cómo el valor de delta influye en el retardo de los paquetes desde el origen hasta el destino. Para un valor de umbral de 20 constante, obtenemos:

	Delta = 100	Delta = 50	Delta = 25
nodos	retardo	retardo	retardo
200	1,12	1,52	2,19
300	1,01	1,27	1,84
400	0,93	1,15	1,59
500	0,91	1,11	1,54
600	0,91	1,11	1,51
700	0,89	1,08	1,46
800	0,88	1,04	1,4

Tabla 1

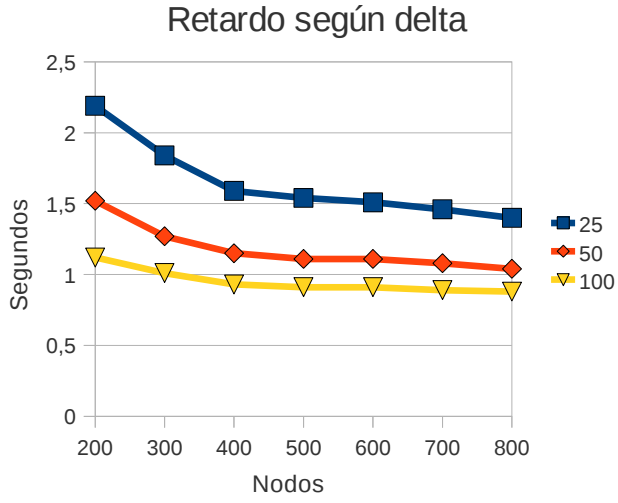

Figura 1

Puede observarse en la Figura y en la Tabla 1 que disminuir el valor de delta mejora el tiempo medio de retardo de entrega de paquetes. Esto es así porque un mayor valor de delta supone un Holding time menor, aunque a cambio de que más nodos retransmitan el mismo paquete, con el consumo de energía que ello supone. También se observa una mejora asintótica de este tiempo al aumentar el número de nodos.

Para observar el consumo de energía, disponemos de la siguiente gráfica:

	Delta = 100	Delta = 50	Delta = 25
nodos	consumo	consumo	consumo
200	4235,55	3474,36	2305,12
300	9642,45	7873,49	5579,36
400	13513,5	11387,2	7355,53
500	18059,7	15087,2	10218,8
600	22511,5	19348,3	13412,9
700	27069,2	23436,4	16664,8
800	33502,2	28448,7	20734,1

Tabla 2

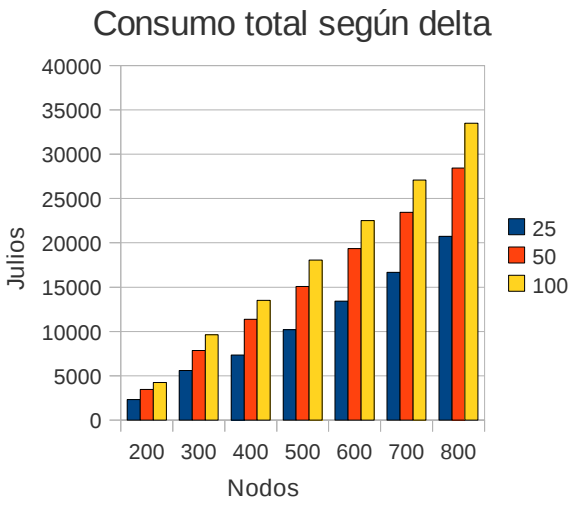

Figura 2

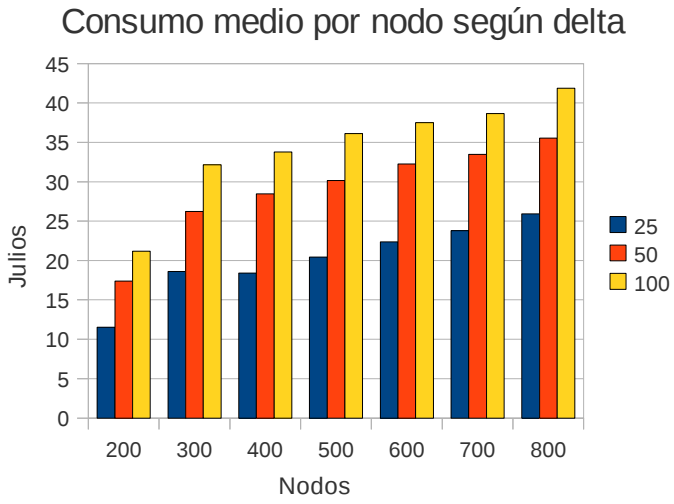

Figura 3

Puede observarse en la Tabla 2 y en las Figuras 2 y 3 como el consumo total de energía crece evidentemente con el número de nodos, y que para la misma cantidad de nodos es mayor cuanto mayor es el valor de delta. Las diferencias entre valores de delta son más pronunciadas que en el paper ya que el consumo de transmisión en nuestra simulación es de 50 W, mientras que en el paper es de 2. Los consumos para el resto de estados no tienen diferencias tan significativas.

A continuación vamos a estudiar cómo influye el valor de umbral en la cantidad de paquetes entregados. Se estudiará para una delta de 25 fija, con un número variable de nodos que pueden moverse hasta una velocidad máxima de 1 m/s. Los resultados obtenidos son los siguientes:

nodos	Umbral = 0 entregados	Umbral = 20 entregados	Umbral = 40 entregados
200	290	279	186
300	300	300	287
400	300	300	300
500	299	300	300
600	300	300	300
700	300	300	300
800	300	300	300

Tabla 3

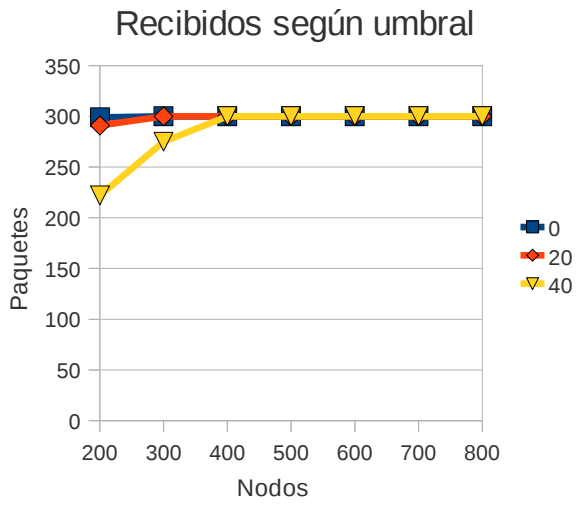

Figura 4

Como puede observarse en la Figura 4, el porcentaje de paquetes entregados tiende al 100% a medida que aumentamos el número de nodos. Un valor de umbral más alto supone una menor entrega de paquetes, ya que en última instancia disminuye el número de nodos que pueden formar parte de una ruta. A cambio, supone un ahorra de energía como vemos en las siguientes gráficas.

Influencia de los parámetros del algoritmo

Estudio del valor delta

Vamos a estudiar cómo cada parámetro configurable del protocolo influye en los resultados de las simulaciones, midiendo principalmente tres factores: el retardo, los paquetes entregados, y el consumo de energía.

Empezaremos por estudiar el valor de delta. Para el escenario descrito arriba, y con un valor constante de umbral de 20, obtenemos los siguientes datos enviando 300 paquetes de 50 bytes, uno por segundo a 10kbps para el modo normal.

nodos	delta	consumo	retardo	recibidos	consumo / nodo
200	25	2305,12	2,19	279	11,526
300	25	5579,36	1,84	300	18,598
400	25	7355,53	1,59	300	18,389
500	25	10218,8	1,54	300	20,438
600	25	13412,9	1,51	300	22,355
700	25	16664,8	1,46	300	23,807
800	25	20734,1	1,4	300	25,918
200	50	3474,36	1,52	291	17,372
300	50	7873,49	1,27	300	26,245
400	50	11387,2	1,15	300	28,468
500	50	15087,2	1,11	300	30,174
600	50	19348,3	1,11	300	32,247
700	50	23436,4	1,08	300	33,481
800	50	28448,7	1,04	300	35,561
200	100	4235,55	1,12	299	21,178
300	100	9642,45	1,01	300	32,142
400	100	13513,5	0,93	300	33,784
500	100	18059,7	0,91	300	36,119
600	100	22511,5	0,91	300	37,519
700	100	27069,2	0,89	300	38,670
800	100	33502,2	0,88	300	41,878

Tabla 4

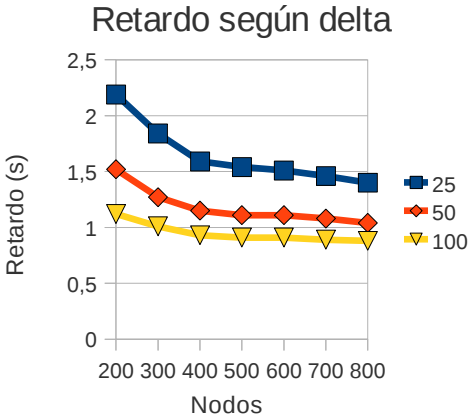

Figura 5

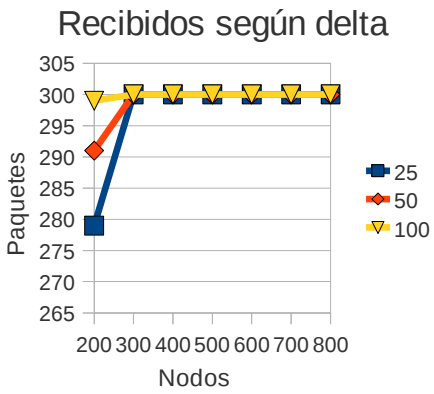

Figura 6

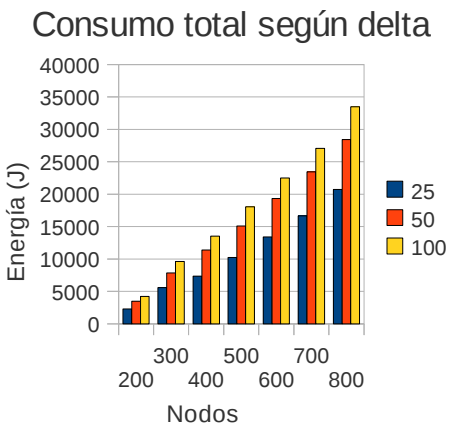

Figura 7

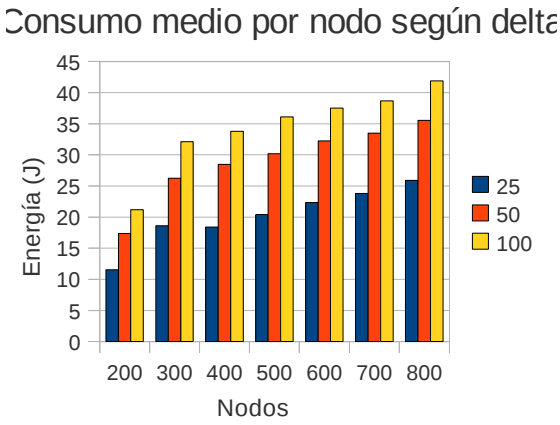

Figura 8

Podemos observar claramente en la Figura 5 que el valor de delta tiene una influencia directa sobre el retardo medio que sufren los paquetes desde un nodo de origen hasta los sumideros. A medida que aumentamos el valor de delta, disminuye el tiempo de retardo. Esto es así porque cómo vemos en la ecuación de cálculo del holding time, delta tiene una influencia directa en el tiempo de espera: a mayor delta, menor tiempo de espera.

Sin embargo, aumentar el valor de delta tiene el inconveniente de aumentar también el consumo de energía, tal como vemos en las figuras 7 y 8. Esto se debe a que al tener un menor tiempo de espera, más nodos retransmiten el mismo paquete, aumentando el consumo de energía.

En la figura 6 podemos ver que para los parámetros utilizados, por encima de los 300 nodos, la entrega de paquetes se sitúa en el 100%. Por debajo de este valor, las diferencias son pequeñas, pero dan a entender que un valor de delta menor supone una menor entrega de paquetes. Desarrollaremos más tests para estudiar como se comporta el protocolo en situaciones extremas.

En general, también podemos afirmar a la vista de los resultados, que un mayor número de nodos supone un menor retardo, que decrece asintóticamente, y un mayor consumo, que crece asintóticamente para el consumo medio por nodo.

Estudio del valor umbral

A continuación, estudiaremos cómo influencia el valor de umbral en los resultados de las simulaciones. Los parámetros serán los mismos que en el anterior apartado, pero manteniendo un valor constante de delta = 50.

nodos	umbral	consumo	retardo	recibidos	consumo/nodo
200	0	4394,34	1,48	299	21,972
300	0	9476,18	1,27	300	31,587
400	0	12957,7	1,15	300	32,394
500	0	16871,3	1,11	300	33,743
600	0	21721,6	1,12	300	36,203
700	0	25987,5	1,08	300	37,125
800	0	30762,6	1,05	300	38,453
200	20	3474,36	1,52	291	17,372
300	20	7873,49	1,27	300	26,245
400	20	11387,2	1,15	300	28,468
500	20	15087,2	1,11	300	30,174
600	20	19348,3	1,11	300	32,247
700	20	23436,4	1,08	300	33,481
800	20	28448,7	1,04	300	35,561
200	40	2290,46	1,4	222	11,452
300	40	4998,69	1,28	275	16,662
400	40	8864,21	1,16	300	22,161
500	40	11954,7	1,11	300	23,909
600	40	15472,7	1,1	300	25,788
700	40	19741,2	1,08	300	28,202
800	40	24303,2	1,03	300	30,379

Tabla 5

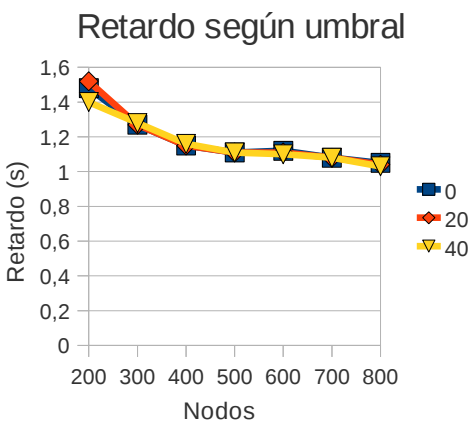

Figura 9

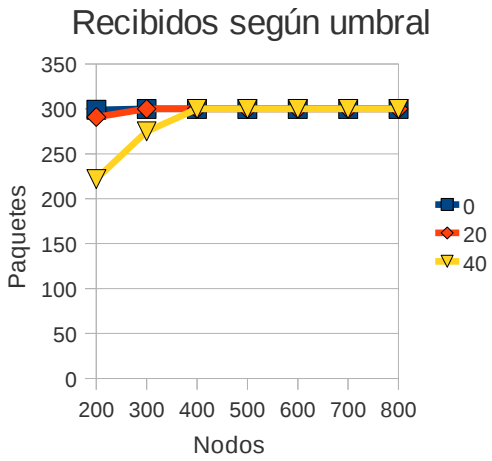

Figura 10

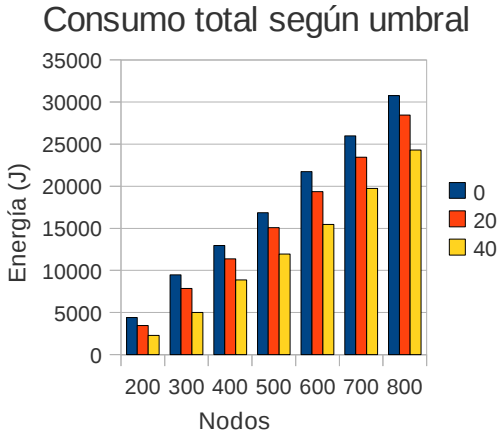

Figura 11

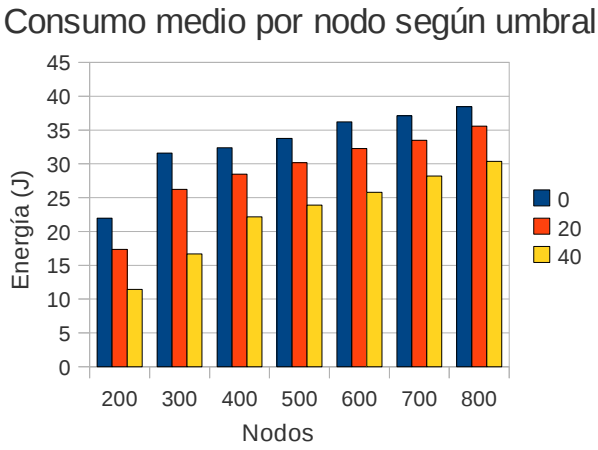

Figura 12

En la figura 9 vemos cómo el valor de umbral tiene muy poca influencia en el retardo de los paquetes. Se debe a que este valor influye en el número de nodos que son candidatos a formar una ruta de entrega, descartando a los nodos que queden demasiado cerca del salto previo, por lo que los candidatos que antes reenvían los paquetes no quedan descartados, lo que hace el retardo permanezca prácticamente igual.

Sin embargo, como podemos observar en la figura 10, a medida que aumentamos el valor de umbral, disminuye la cantidad de paquetes entregados, pues más rutas pueden quedar incompletas al disminuir el número de nodos candidatos. Además, a medida que disminuye el número de nodos, la cantidad de paquetes entregados cae bruscamente.

Por tanto, la ventaja evidente de un valor grande de umbral, es que disminuye la energía consumida, prolongando la vida de los nodos. De las figuras 11 y 12 se desprende que, si bien el consumo medio por nodo crece asintóticamente, es considerablemente menor para valores mayores de umbral.

Estudio de la velocidad de los nodos

Vamos a estudiar cómo influye la velocidad de desplazamiento de los nodos en los resultados de la simulaciones. Seguiremos la misma distribución que en el artículo de referencia, 3 velocidades: 1, 5 y 10 m/s. Con ella obtendremos:

nodos	velocidad	consumido	cons. Medio	etransmisiones	retardo	recibidos
200	1	3050	15,250	12024	2,33	290
300	1	6957	23,190	27523	1,91	300
400	1	9134	22,835	31416	1,64	300
500	1	12573	25,146	38239	1,58	299
600	1	15368	25,613	46355	1,52	300
700	1	19290	27,557	54567	1,5	300
800	1	23152	28,940	64451	1,45	300
200	5	3009	15,045	9508	2,42	220
300	5	6460	21,533	22549	2,24	275
400	5	9619	24,048	32887	1,76	296
500	5	13171	26,342	42548	1,68	300
600	5	17197	28,662	50725	1,62	300
700	5	20138	28,769	60357	1,56	300
800	5	24290	30,363	69101	1,49	300
200	10	2690	13,450	5161	2,4	137
300	10	5788	19,293	17135	2,1	214
400	10	8808	22,020	27232	1,84	256
500	10	13544	27,088	42648	1,82	294
600	10	17198	28,663	50755	1,68	298
700	10	20810	29,729	60020	1,62	299

Tabla 6

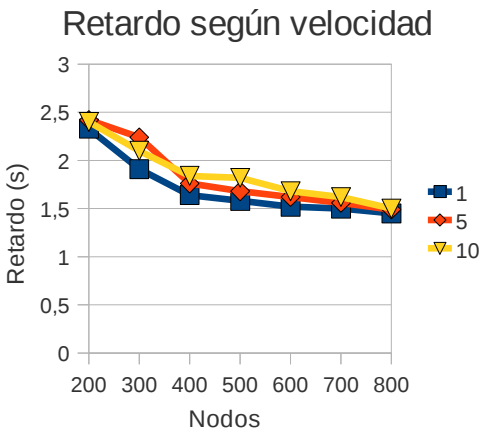

Figura 13

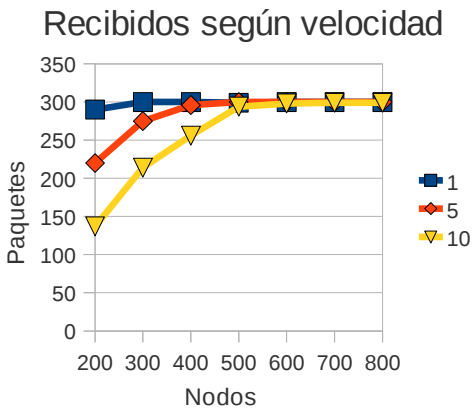

Figura 14

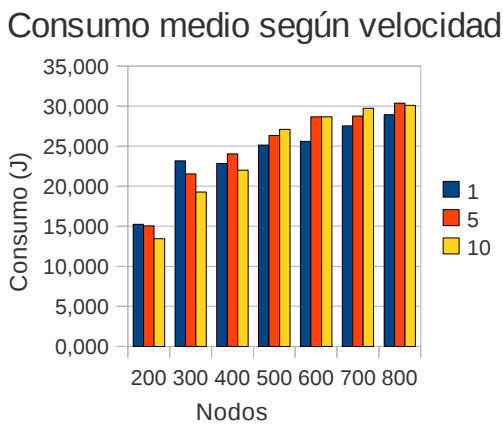

Figura 15

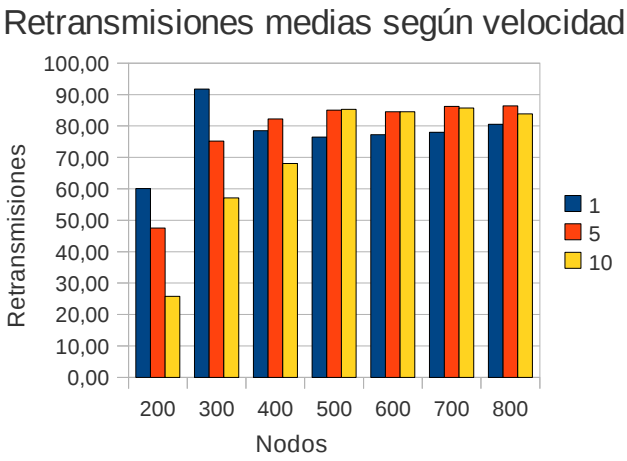

Figura 16

Como era de esperar, la velocidad de desplazamiento de los nodos, tiene una influencia prácticamente nula sobre el retardo con el que los paquetes llegan hasta los sumideros, tal como muestra la Figura 13. Sin embargo, cuanto mayor es la velocidad de desplazamiento, más paquetes se pierden, debido a que las rutas quedan rotas sin encontrar nodos que las completen, como en la Figura 14.

En las Figuras 15 y 16, se observa que la influencia sobre el consumo y el número de retransmisiones es mayor cuanto menor es el número de nodos, pues la red, al volverse más frágil, más nodos acaban retransmitiendo los mismo paquetes para completar la ruta. A medida que aumentan los nodos, y por tanto más probable es que exista una ruta, los resultados tienen a converger a los mismos valores, habiendo una diferencia muy pequeña para las distintas velocidades.

Estudio del modo extendido

Estudiaremos también el rendimiento del modo extendido implementado, en el que se tiene en cuenta el tiempo de transmisión para calcular el Holding Time, y veremos cómo se comporta para la entrega de paquetes y retardo de éstos. Este modo se vuelve interesante cuando la velocidad de transmisión es muy baja, de forma que el tiempo de transmisión se vuelve muy grande, llegando a superar al de propagación. En las siguientes simulaciones se ha utilizado una velocidad de 1500 bps. Los parámetros de energía con que se configuran los nodos son de gran importancia en estas simulaciones. Cada nodo intermedio tiene una batería con 500 J de energía, y unos consumos de 50 W para transmisión, y 158 mW para recepción e idle, que son los que vienen por defecto para el micromódem del ns-3. Los nodos fuente y sumideros tienen baterías con capacidad sobrada para que no se agoten.

En un primer estudio preliminar, obtenemos:

Recibidos según modo

	Delta = 25, umbral = 0		Delta = 50, Umbral = 0		Delta = 100, umbral = 0	
nodos	normal	extendido	normal	extendido	normal	extendido
200	45	40	44	51	45	44
300	69	78	69	73	70	84
400	79	106	85	98	70	90
500	87	107	92	107	76	91
600	90	123	93	93	78	92
700	88	125	100	101	83	102
800	95	126	104	99	82	109

Tabla 7

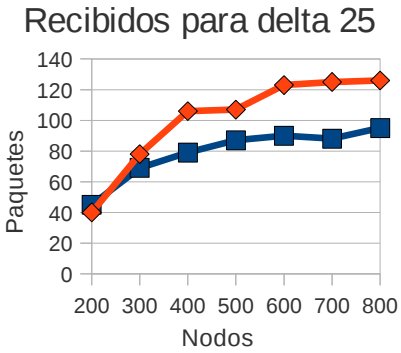

Figura 17

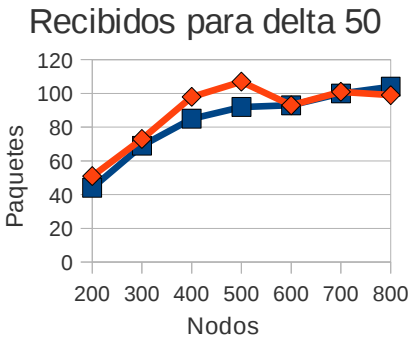

Figura 18

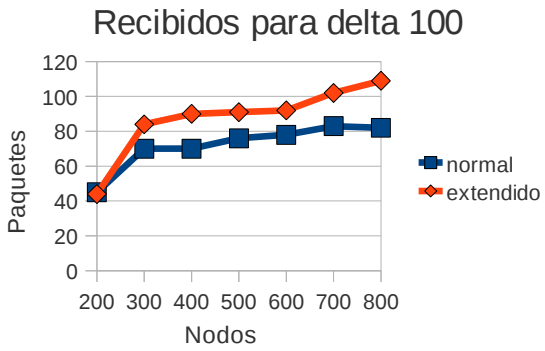

Figura 19

En las Figuras 17 a 19, se observa que, como cabría esperar, ofrece unas mejores tasas de entrega de paquete para la mayoría de casos. Sin embargo, veamos lo que ocurre cuando estudiamos el retardo:

Retardo según modo

	Delta = 25, umbral = 0		Delta = 50, Umbral = 0		Delta = 100, umbral = 0	
nodos	normal	extendido	normal	extendido	normal	extendido
200	5,503	8,685	4,191	5,99	3,276	4,814
300	5,413	9,767	4,136	6,195	3,732	4,986
400	5,663	10,338	4,4	6,892	3,659	5,383
500	6,361	11,388	4,534	6,851	3,746	5,324
600	5,908	12,432	4,459	6,819	3,408	5,144
700	6,117	12,048	4,521	6,725	3,687	5,542
800	6,235	11,758	4,509	7,233	3,455	5,857

Tabla 8

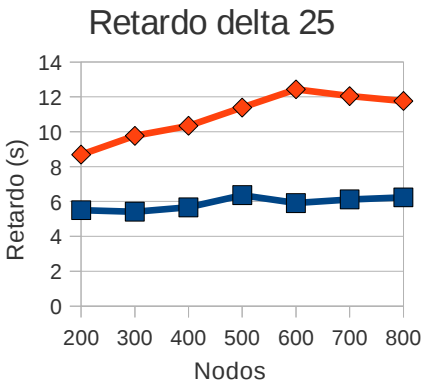

Figura 20

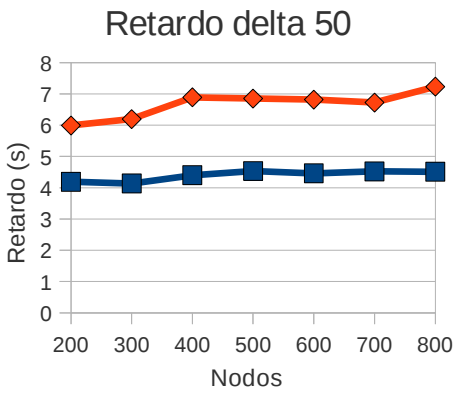

Figura 21

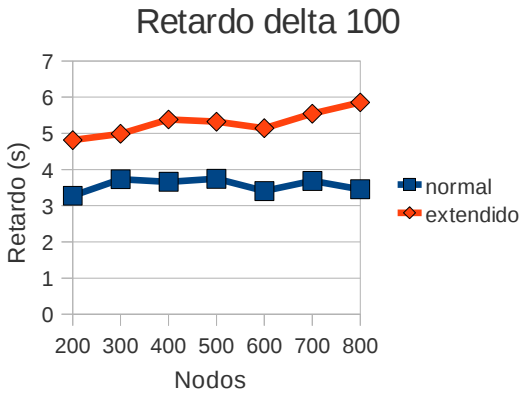

Figura 22

En las Figuras 20 a 22 se observa la penalización que tiene el modo extendido: un retardo mayor, ya que tiene en cuenta otro tiempo más para calcular el tiempo de espera. Quedaría pues a elección del criterio de si se prefiere una mayor tasa de entregas o un mejor tiempo de retardo.

Procedemos a estudiar cómo afecta la velocidad de transmisión al rendimiento de entrega de paquetes para ambos modos, por lo que ejecutaremos una serie de simulaciones en que se variará el datarate. Utilizaremos 3 escenarios distintos, para ver el comportamiento con una población diferente de nodos, y a distintas velocidades de desplazamiento. A continuación los primeros resultados obtenidos:

	Paquetes entregados de un total de 300 para delta = 50					
	200 nodos	5 m/s	200 nodos	1 m/s	300 nodos	5 m/s
datarate	normal	extendido	normal	extendido	normal	extendido
1000	43	43	25	26	66	67
2000	91	86	63	48	115	116
3000	136	126	73	86	160	183
4000	177	163	110	129	206	216
5000	206	207	156	140	232	241
6000	215	202	182	163	269	271
7000	217	223	207	200	281	276
8000	232	225	237	232	280	273
9000	236	229	257	232	280	277
10000	237	230	251	272	280	280

Tabla 9

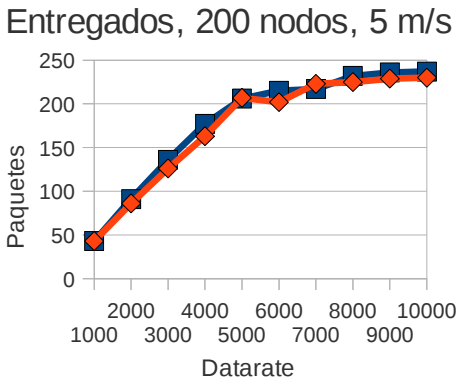

Figura 23

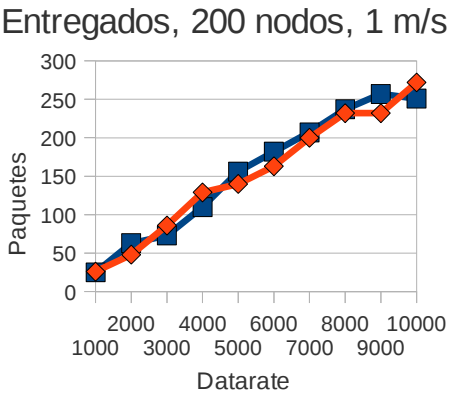

Figura 24

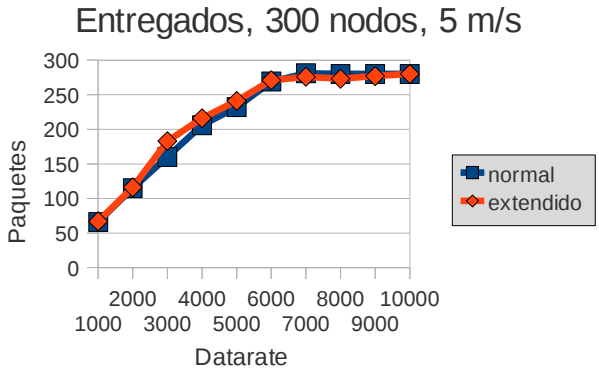

Figura 25

Al contrario de lo que podríamos pensar, se observa que en las figuras 23 a 25, las diferencias son mínimas, por no decir que el modo normal ofrece un mejor rendimiento en la mayoría de casos para la entrega de paquetes, cuando cabría pensar que sería al revés.

Por ello, se decide ejecutar nuevas simulaciones para un valor de delta distinto, pues en los resultados preliminares anteriores a 1500 bps, el valor de 50 usado, era el que menor diferencia ofrecía, por lo que se opta por repetir los tests con un valor de 25:

	Paquetes entregado para un total de 300 para delta 25					
	200 nodos	5 m/s	200 nodos	1 m/s	300 nodos	5 m/s
datarate	normal	extendido	normal	extendido	normal	extendido
1000	43	40	31	30	65	57
2000	90	86	62	59	127	132
3000	120	112	93	88	174	190
4000	158	151	121	127	209	238
5000	181	159	127	188	228	238
6000	192	176	157	238	259	254
7000	208	185	176	203	277	259
8000	208	192	228	244	275	260
9000	221	199	250	242	275	261
10000	220	200	272	267	275	266

Tabla 10

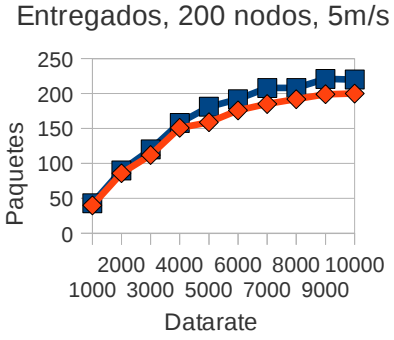

Figura 26

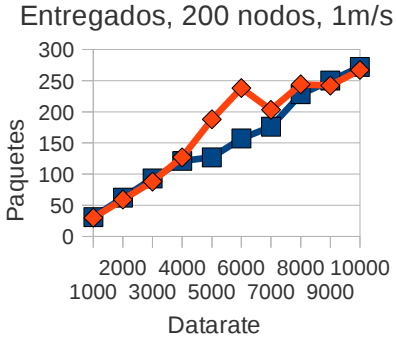

Figura 27

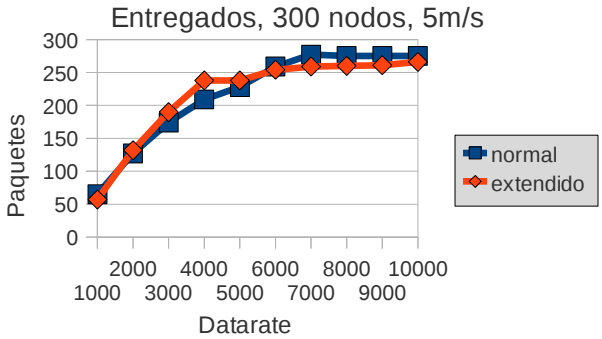

Figura 28

Aunque no muestran una superioridad del modo normal sobre el extendido como insinuaban los anteriores resultados, no se puede tampoco definir una preferencia clara por un modo u otro basándose únicamente en la tasa de paquetes entregados, tal como se observa en las Figuras 26 a 28. Por tanto, el retardo inclinaría la balanza a favor del modo normal.

Si observamos los resultados preliminares, vemos que para una delta de 25 y con 400 nodos, el modo extendido entrega más paquetes que el normal. Vamos a probar a variar el datarate para estos parámetros, obteniendo los siguientes resultados, mostrados en la Tabla 11 y la Figura 29:

Paquetes recibidos de 300 enviados para delta 25 y 400 nodos		
datarate	normal	extendido
1000	72	91
2000	130	156
3000	178	223
4000	230	272
5000	285	288
6000	294	288
7000	296	285
8000	296	279
9000	295	294
10000	296	294

Tabla 11

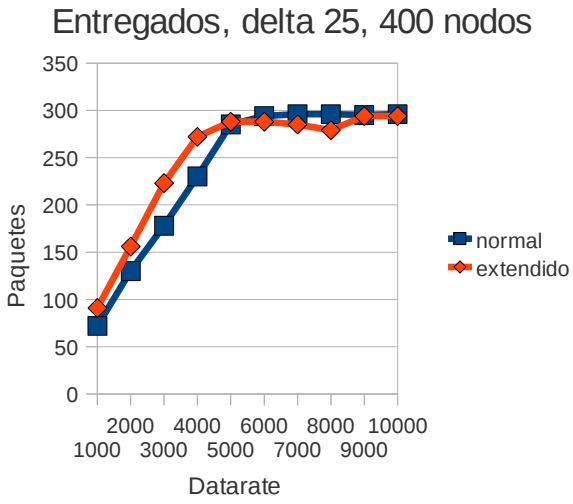

Figura 29

Vamos a intentar ver un caso de uso en el que el empleo del modo extendido tenga un mejor rendimiento claro sobre el normal. Para ello probaremos con un datarate mucho más bajo. Los resultados se muestran en la Tabla 12 y la Figura 30.

Paquetes recibidos de 300 enviados para delta 25 y 400 nodos		
datarate	normal	extendido
100	9	14
200	17	20
300	21	32
400	30	35
500	42	44
600	50	45
700	48	62
800	61	75
900	63	77
1000	72	91

Tabla 12

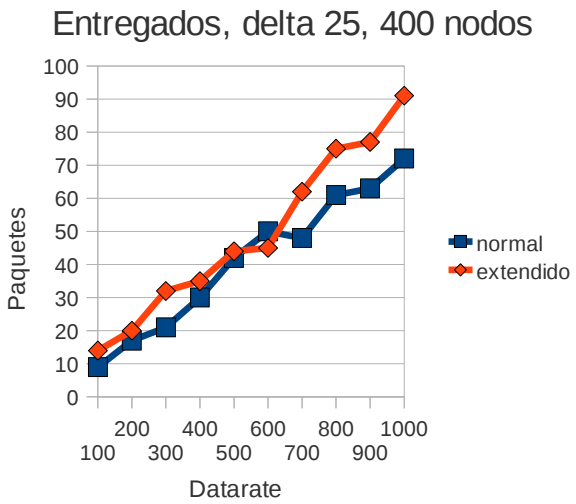

Figura 30

Puede observarse cómo para valores muy bajos para la velocidad de transmisión, el modo extendido presenta un mejor rendimiento.

Sin embargo, si estudiamos en profundidad por qué se entregan más paquetes con el modo extendido, veremos que es debido al consumo de energía por parte de los nodos, ya que es menor, lo que permite a más nodos sobrevivir más tiempo para entregar más paquetes.

Para la Tabla 13, veamos cómo evoluciona el consumo de energía, los nodos agotados, y el número de retransmisiones para los mismo parámetros de la simulación:

Resultados para delta 25 y 400 nodos

	consumo energía		nodos agotados		retransmisiones	
datarate	normal	extendido	normal	extendido	normal	extendido
100	196148	192237	342	336	1012	976
200	194665	185621	330	291	1729	1577
300	195842	183522	315	270	2457	2205
400	190981	181721	311	260	3368	2914
500	196505	187682	319	260	4133	3701
600	193975	183918	315	233	4850	4337
700	193491	183632	316	251	5696	5107
800	192927	185063	316	273	6559	5982
900	191046	180155	314	234	7275	6393
1000	189857	179297	310	239	7983	7129

Tabla 13

De donde obtenemos las Figuras 31 a 33:

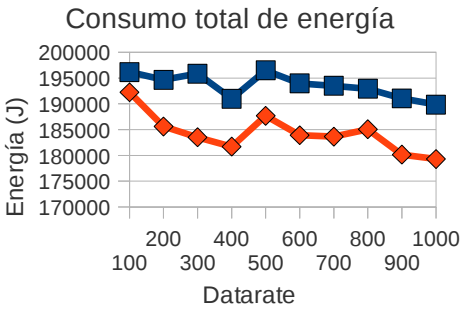

Figura 31

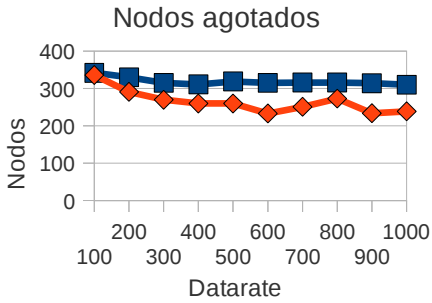

Figura 32

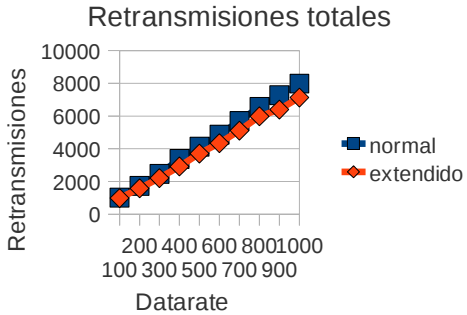

Figura 33

Estas figuras nos dan la clave del mayor número de paquetes entregados: que el menor consumo energético del modo extendido permite que más nodos sobrevivan con el paso del tiempo y por tanto sean candidatos a formar rutas. En el análisis de este modo hemos estado utilizando los parámetros de consumo por defecto del micromódem propuesto por el ns-3, y una batería de 500 J, lo que da lugar a que una gran cantidad de nodos agoten su energía. Sin embargo, si repetimos algunos de estos estudios con los parámetros de consumo propuestos en el artículo, manteniendo la misma batería, obtendremos la Tabla 14:

	Paquetes recibidos		Consumo energía		Retransmisiones	
datarate	normal	extendido	normal	extendido	normal	extendido
100	21	18	40349	30519	2759	1773
200	40	27	36829	27552	4694	2765
300	52	41	36313	26768	6798	4068
400	70	53	35900	26249	9012	4895
500	88	75	33466	27484	10189	6646
600	113	82	34645	26521	12696	7519
700	116	92	33032	25170	13986	8387
800	109	101	30589	23640	13587	9143
900	116	109	29508	24062	14003	10177
1000	143	120	30789	23075	16670	10626

Tabla 14

En ninguna de estas simulaciones se ha agotado ningún nodo. Podemos dibujar las Figuras 34 a 36:

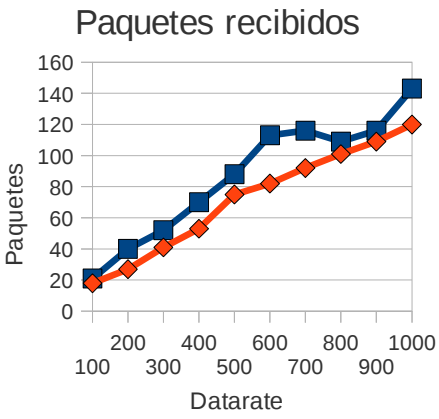

Figura 34

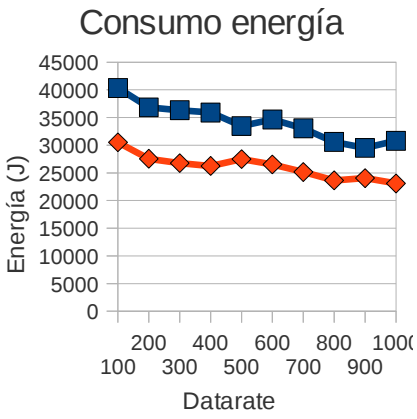

Figura 35

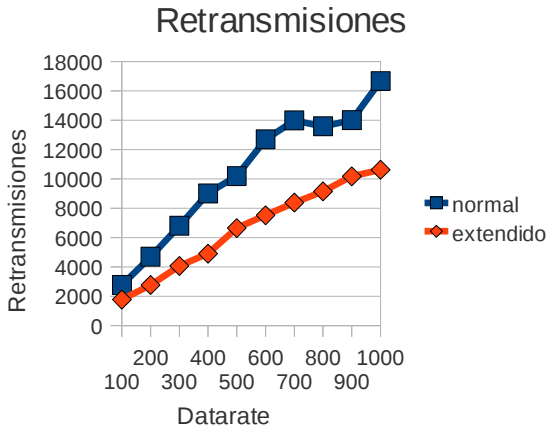

Figura 36

De estos datos se desprende que cuando no se agotan la energía de los nodos, la cantidad de paquetes entregados es mayor para el modo normal. Por tanto, las supuesta superioridad en la entrega de paquetes del modo extendido únicamente viene dada por el hecho de que más nodos sobreviven a largo plazo, lo que hace que en situaciones críticas, cuando un importante porcentaje de nodos ha muerto, el modo extendido pueda seguir entregando paquetes.

5. Conclusiones y trabajo futuro

En el estudio del protocolo DBR hemos llegado a confirmar los resultados presentados en el artículo.

En primer lugar, hemos comprobado la influencia que tiene el valor de la ecuación delta, que a medida que aumenta, disminuye el retardo de los paquetes, a costa de aumentar el consumo de energía. También hemos comprobado cómo influencia el valor de umbral a la cantidad de paquetes entregados, disminuyendo el número de recibidos para valores mayores, pero presentando un mejor consumo energético.

También hemos tenido la oportunidad de presentar una posible mejora sobre el protocolo original y estudiar si realmente supone una diferencia positiva. La idea propuesta consistía en tener en cuenta el tiempo de transmisión para el cálculo del holding time, pues en este tipo de redes los datarates son bastante bajos, por lo que para ciertas configuraciones se consideraba que era un valor que no se podía obviar. Se preveía que tuviera un impacto positivo en la cantidad de paquetes entregados, al disminuir el número de posibles colisiones, a costa de aumentar el retardo de los paquetes.

Sin embargo, en las simulaciones hemos comprobado que el impacto sobre los paquetes entregados es nulo o peor que para el modo normal en la mayoría de los casos de uso. EL retardo, como se había previsto, es mayor para este modo. La única ventaja que se obtenía era en el tiempo de vida de la red, ya que menos nodos retransmiten paquetes, lo que prolonga la vida de éstos, pudiendo seguir entregando paquetes cuando en el modo normal no quedan suficientes para crear una ruta válida. Por tanto, el modo extendido, aunque siempre presenta un menor consumo energético, también presenta siempre un retardo mayor. En cuanto a los paquetes entregados, dependen únicamente de los nodos vivos, y por regla general, para la misma cantidad de nodos vivos, el modo normal entrega más paquetes.

De esta forma, se recomendaría la utilización de esta alternativa en aquellos casos en que las restricciones energéticas de los nodos supongan una dificultad a vencer, en donde los nodos agoten rápidamente su energía o la red ofrezca un tiempo corto de vida.

Como trabajo futuro se propone adaptar el protocolo MAC CW para que trabaje en conjunto con DBR. En la implementación actual, cuando se entrega un paquete a la capa MAC para su envío, el protocolo de enrutamiento de la capa superior pierde todo control sobre él. Nuestra propuesta consistiría en adaptar la MAC para que al final de la ventana de contención, justo antes de entregar en paquete al medio físico, haga una última comprobación preguntando a DBR si un paquete similar llegó durante la ventana de contención y por tanto abortar el envío en el último momento, con el consiguiente ahorro de energía y posibles colisiones.

6. Bibliografía

[1] DBR: Depth-Based Routing for Underwater Sensor Networks , Hai Yan, Zhijie Jerry Shi, and Jun-Hong Cui

[2] Robot sub reached deepest ocean, http://news.bbc.co.uk/2/hi/science/nature/8080324.stm

[3] Wikipedia, Sonar, http://es.wikipedia.org/wiki/Sonar

[4] An introduction to underwater acoustics: principles and applications, Xavier Lurton

[5] A review of routing protocols for mobile ad hoc networks , Mehran Abolhasan, Tadeusz Wysocki, Eryk Dutkiewicz

[6] Homepage of ns-3: http://www.nsnam.org/

[7] P. Xie, J.-H. Cui, and L. Lao. Vbf: Vector-based forwarding protocol for underwater sensor networks.